1. Docker Laravel offline (20 minutes)

Objectif

Configurer un environnement Docker pour Laravel sans connexion internet

Prérequis

- Docker Desktop installé
- Images Docker téléchargées à l'avance

Configuration

Créer un fichier `docker-compose.yml` :

```
version: '3.8'

services:
  app:
    image: php:8.2-fpm
    container_name: laravel_app
    volumes:
      - ./:/var/www/html
    networks:
      - laravel

  nginx:
    image: nginx:alpine
    container_name: laravel_nginx
    ports:
      - "8000:80"
    volumes:
      - ./:/var/www/html
      - ./nginx.conf:/etc/nginx/conf.d/default.conf
    networks:
      - laravel

  mysql:
    image: mysql:8.0
    container_name: laravel_mysql
```

```
    environment:
      MYSQL_DATABASE: laravel
      MYSQL_ROOT_PASSWORD: root
      MYSQL_PASSWORD: secret
      MYSQL_USER: laravel
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - laravel

networks:
  laravel:
    driver: bridge

volumes:
  mysql_data:
```

Démarrer les conteneurs :

```
docker-compose up -d
```

2. Installation Composer et Laravel (15 minutes)

Installation de Composer

Si Composer n'est pas installé :

```
# Linux/Mac
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer

# Windows : télécharger l'installeur depuis getcomposer.org
```

Vérification

```
composer --version
```

Installation de Laravel Installer

```
composer global require laravel/installer
```

3. laravel new (10 minutes)

Création d'un nouveau projet

```
# Méthode 1 : avec Laravel Installer
laravel new mon-projet

# Méthode 2 : avec Composer
composer create-project laravel/laravel mon-projet
```

Accéder au projet

```
cd mon-projet
```

4. Configuration .env (15 minutes)

Le fichier .env

Copier le fichier d'exemple :

```
cp .env.example .env
```

Configurations importantes

```
# Application
APP_NAME="Mon Application Laravel"
APP_ENV=local
APP_KEY=
APP_DEBUG=true
```

```
APP_URL=http://localhost:8000

# Database
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=laravel
DB_PASSWORD=secret
```

Générer la clé d'application

```
php artisan key:generate
```

5. Découverte de la structure (20 minutes)

Exploration guidée

- [ ] Explorer le dossier `app/` et identifier les Controllers et Models
- [ ] Ouvrir `routes/web.php` et examiner les routes par défaut
- [ ] Regarder `resources/views/welcome.blade.php`
- [ ] Explorer `database/migrations/`
- [ ] Vérifier `config/app.php` et `config/database.php`

Lancer le serveur de développement

```
php artisan serve
```

Accéder à `http://localhost:8000` dans le navigateur

6. Création de routes simples (20 minutes)

Éditer routes/web.php

```
<?php
```

```php
use Illuminate\Support\Facades\Route;

// Route de base
Route::get('/', function () {
    return view('welcome');
});

// Route simple avec texte
Route::get('/hello', function () {
    return 'Bonjour Laravel!';
});

// Route avec paramètre
Route::get('/user/{name}', function ($name) {
    return "Bonjour, $name!";
});

// Route avec paramètre optionnel
Route::get('/user/{name?}', function ($name = 'Invité') {
    return "Bonjour, $name!";
});

// Route retournant une view
Route::get('/about', function () {
    return view('about');
});

// Route avec données passées à la view
Route::get('/contact', function () {
    $email = 'contact@example.com';
    return view('contact', ['email' => $email]);
});
```

Créer une view simple

Créer `resources/views/about.blade.php` :

```
<!DOCTYPE html>
<html>
<head>
    <title>À propos</title>
</head>
<body>
    <h1>À propos de nous</h1>
    <p>Bienvenue sur notre site Laravel!</p>
</body>
</html>
```

Tester les routes

- [ ] Tester `http://localhost:8000/hello`
- [ ] Tester `http://localhost:8000/user/Jean`
- [ ] Tester `http://localhost:8000/about`

7. Migration categories (20 minutes)

Créer une migration

```
php artisan make:migration create_categories_table
```

Éditer la migration

Fichier : `database/migrations/YYYY_MM_DD_HHMMSS_create_categories_table.php`

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
```

```
     */
    public function up(): void
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('slug')->unique();
            $table->text('description')->nullable();
            $table->boolean('is_active')->default(true);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('categories');
    }
};
```

Exécuter la migration

```
php artisan migrate
```

Commandes utiles

```
# Voir le statut des migrations
php artisan migrate:status

# Annuler la dernière migration
php artisan migrate:rollback

# Réinitialiser et relancer toutes les migrations
php artisan migrate:fresh
```

8. Seeder catégories (20 minutes)

Créer un modèle Category

```
php artisan make:model Category
```

Éditer le modèle

Fichier : `app/Models/Category.php`

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'slug',
        'description',
        'is_active'
    ];

    protected $casts = [
        'is_active' => 'boolean',
    ];
}
```

Créer un seeder

```
php artisan make:seeder CategorySeeder
```

Éditer le seeder

Fichier : `database/seeders/CategorySeeder.php`

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use App\Models\Category;

class CategorySeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $categories = [
            [
                'name' => 'Technologie',
                'slug' => 'technologie',
                'description' => 'Articles sur la technologie et l\'innovation',
                'is_active' => true
            ],
            [
                'name' => 'Science',
                'slug' => 'science',
                'description' => 'Découvertes scientifiques et recherches',
                'is_active' => true
            ],
            [
                'name' => 'Sport',
                'slug' => 'sport',
                'description' => 'Actualités sportives',
                'is_active' => true
            ],
            [
                'name' => 'Culture',
```

```
            'slug' => 'culture',
            'description' => 'Arts, musique et culture',
            'is_active' => true
        ],
        [
            'name' => 'Économie',
            'slug' => 'economie',
            'description' => 'Actualités économiques et financières',
            'is_active' => true
        ]
    ];

    foreach ($categories as $category) {
        Category::create($category);
    }
  }
}
```

Enregistrer le seeder

Éditer `database/seeders/DatabaseSeeder.php` :

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
  /**
   * Seed the application's database.
   */
  public function run(): void
  {
    $this->call([
      CategorySeeder::class,
    ]);
```

```
    }
}
```

Exécuter le seeder

```
# Exécuter tous les seeders
php artisan db:seed

# Exécuter un seeder spécifique
php artisan db:seed --class=CategorySeeder

# Réinitialiser la base et lancer les seeders
php artisan migrate:fresh --seed
```

Vérifier les données

Créer une route de test dans `routes/web.php` :

```
use App\Models\Category;

Route::get('/categories', function () {
    $categories = Category::all();
    return view('categories', ['categories' => $categories]);
});
```

Créer `resources/views/categories.blade.php` :

```
<!DOCTYPE html>
<html>
<head>
    <title>Catégories</title>
</head>
<body>
    <h1>Liste des catégories</h1>
    <ul>
        @foreach($categories as $category)
            <li>
                <strong>{{ $category->name }}</strong> ({{ $category->slug }})
```

```
            <br>
            {{ $category->description }}
        </li>
    @endforeach
</ul>
</body>
</html>
```

Exercices supplémentaires

- [ ] Créer une route pour afficher une catégorie spécifique par son slug
- [ ] Ajouter une colonne `color` à la table categories via une nouvelle migration
- [ ] Créer un controller `CategoryController` avec la méthode `index()`
- [ ] Modifier le seeder pour ajouter 3 catégories supplémentaires